Structured Sparse Principal Component Analysis

Rodolphe Jenatton, Guillaume Obozinski, Francis Bach

Structured Sparse Principal Component Analysis

Rodolphe Jenatton[1] rodolphe.jenatton@inria.fr
Guillaume Obozinski[1] guillaume.obozinski@inria.fr
Francis Bach[1] francis.bach@inria.fr
[1]*INRIA - WILLOW Project-team,*
Laboratoire d'Informatique de l'Ecole Normale Supérieure (INRIA/ENS/CNRS UMR 8548),
23, avenue d'Italie, 75214 Paris. France

September 8, 2009

Abstract

We present an extension of sparse PCA, or sparse dictionary learning, where the sparsity patterns of all dictionary elements are structured and constrained to belong to a prespecified set of shapes. This *structured sparse PCA* is based on a structured regularization recently introduced by [1]. While classical sparse priors only deal with *cardinality*, the regularization we use encodes higher-order information about the data. We propose an efficient and simple optimization procedure to solve this problem. Experiments with two practical tasks, face recognition and the study of the dynamics of a protein complex, demonstrate the benefits of the proposed structured approach over unstructured approaches.

1 Introduction

Principal component analysis (PCA) is an essential tool for data analysis and unsupervised dimensionality reduction, whose goal is to find, among linear combinations of the data variables, a sequence of orthogonal factors that most efficiently explain the variance of the observations.

One of its main shortcomings is that, even if PCA finds a small number of important factors, the factor themselves typically involve all original variables. In the last decade, several alternatives to PCA which find sparse and potentially interpretable factors have been proposed, notably nonnegative matrix factorization (NMF) [2] and sparse PCA (SPCA) [3, 4, 5].

However, in many applications, only constraining the size of the factors does not seem appropriate because the considered factors are not only expected to be sparse but also to have a certain structure. In fact, the popularity of NMF for face image analysis owes essentially to the fact that the method happens to retrieve sets of variables that are localized on the face and capture some features or parts of the face which seem intuitively meaningful given our a priori. We might therefore gain in the quality of the factors induced by enforcing directly this a priori in the matrix factorization constraints. More generally, it is desirable to encode higher-order information about the supports that reflects the *structure* of the data. For example, in computer vision, features associated to the pixels of an image are naturally organized on a grid and the supports of factors explaining the variability of images could be expected to be localized, connected or have some other regularity with respect to the grid. Similarly, in genomics, factors explaining the gene expression patterns

observed on a microarray could be expected to involve groups of genes corresponding to biological pathways or set of genes that are neighbors in a protein-protein interaction network.

Recent research on structured sparsity [6, 7, 1] has highlighted the benefit of exploiting such structure for variable selection and prediction in the context of regression and classification. In particular, [1] shows that, given any intersection-closed family of patterns $\mathcal{P}$ of variables, such as all the rectangles on a 2-dimensional grid of variables, it is possible to build an ad hoc regularization norm Ω that enforces that the support of the solution of the least-squares regression regularized by Ω belongs to the family $\mathcal{P}$.

Capitalizing on these results, we aim in this paper to go beyond sparse PCA and propose *structured sparse PCA* (SSPCA), which explains the variance of the data by factors that are not only sparse but also respect some a priori structural constraints deemed relevant to model the data at hand. We show how slight variants of the regularization term of [1] can be used successfully to yield a structured and sparse formulation of principal component analysis for which we propose a simple and efficient optimization scheme.

The rest of the paper is organized as follows: Section 2 introduces the SSPCA problem in the dictionary learning framework, summarizes the regularization considered in [1] and its essential properties, and presents some simple variants which are more effective in the context of PCA. Section 3 is dedicated to our optimization scheme for solving SSPCA. Our experiments in Section 4 illustrate the benefits of our approach through applications to face recognition and the study of the dynamics of protein complexes.

Notation: For any vector y in $\mathbb{R}^p$ and any $\alpha > 0$, we denote by $\|y\|_\alpha = (\sum_{j=1}^p |y_j|^\alpha)^{1/\alpha}$ the (quasi-)norm ℓ_α of y. Similarly, for any rectangular matrix $Y \in \mathbb{R}^{n\times p}$, we denote by $\|Y\|_{\mathrm{F}} = (\sum_{i=1}^n \sum_{j=1}^p Y_{ij}^2)^{1/2}$ its Frobenius norm, where Y_{ij} is the (i,j)-th element of Y. We write Y^j for the j-th column of Y. Given w in $\mathbb{R}^p$ and a subset J of $\{1,\dots,p\}$, w_J denotes the vector in $\mathbb{R}^p$ that has the same entries w_j as w for $j \in J$, and null entries outside of J. In addition, $\mathrm{supp}(w) = \{j \in \{1,\dots,p\}\,;\, w_j \neq 0\}$ is referred to as the *support*, or *nonzero pattern* of the vector $w \in \mathbb{R}^p$. For any finite set A with cardinality $|A|$, we also define the $|A|$-tuple $(y^a)_{a\in A} \in \mathbb{R}^{p\times|A|}$ as the collection of p-dimensional vectors y^a indexed by the elements of A. Furthermore, for two vectors x and y in $\mathbb{R}^p$, we denote by $x \circ y = (x_1y_1,\dots,x_py_p)^\top \in \mathbb{R}^p$ the elementwise product of x and y. Finally, we extend $\frac{a}{b}$ by continuity in zero with $\frac{a}{0} = \infty$ if $a \neq 0$ and 0 otherwise.

2 Problem statement

It is useful to distinguish two conceptually different interpretations of PCA. In terms of *analysis*, PCA sequentially projects the data on subspaces that explain the largest fraction of the variance of the data. In terms of *synthesis*, PCA finds a basis, or orthogonal dictionary, such that all signals observed admit decompositions with low reconstruction error. These two interpretations recover the same basis of principal components for PCA but lead to different formulations for *sparse* PCA. The *analysis* interpretation leads to sequential formulations ([8, 9, 3]) that consider components one at a time and perform a *deflation* of the covariance matrix at each step (see [10]). The *synthesis* interpretation leads to non-convex global formulations ([4, 11, 9, 12]) which estimate simultaneously all principal components, often drop the orthogonality constraints, and are referred to as matrix factorization problems ([13]) in machine learning, and dictionary learning in signal processing.

The approach we propose fits more naturally in the framework of dictionnary learning, whose terminology we now introduce.

2.1 Matrix factorization and dictionary learning

Given a matrix $X \in \mathbb{R}^{n \times p}$ of n rows corresponding to n observations in $\mathbb{R}^p$, the dictionary learning problem is to find a matrix $V \in \mathbb{R}^{p \times r}$, called the *dictionary*, such that each observation can be well approximated by a linear combination of the r columns $(V^k)_{k \in \{1,\dots,r\}}$ of V called the *dictionary elements*. If $U \in \mathbb{R}^{n \times r}$ is the matrix of the linear combination coefficients or *decomposition coefficients*, the matrix product $UV^\top$ is called a decomposition of X.

Learning simultaneously the dictionary V and the decomposition U corresponds to a matrix factorization problem (see [5] and reference therein). As formulated in [14] or [5], it is natural, when learning a decomposition, to penalize or constrain some norms or quasi-norms of U and V, say Ω_u and Ω_v respectively, to encode prior information—typically sparsity—about the decomposition of X. This can be written generally as

$$\min_{U \in \mathbb{R}^{n \times r},\, V \in \mathbb{R}^{p \times r}} \frac{1}{2np} \left\| X - UV^\top \right\|_{\mathrm{F}}^2 + \lambda \sum_{k=1}^{r} \Omega_v(V^k) \qquad \text{s.t.} \qquad \forall k,\, \Omega_u(U^k) \leq 1, \tag{1}$$

where the regularization parameter $\lambda \geq 0$ controls which extent the dictionary is regularized[1]. If we assume that both regularizations Ω_u and Ω_v are convex, problem (1) is convex w.r.t. U for V fixed and vice versa. It is however not *jointly* convex in (U, V).

The formulation of sparse PCA considered in [12] corresponds to a particular instance of this problem, where the dictionary elements are required to be sparse (without the orthogonality constraint $V^\top V = I$). This can be achieved by penalizing the columns of V by a sparsity-inducing norm, e.g., the ℓ_1 norm, $\Omega_v(V^k) = \left\| V^k \right\|_1$. In the next section we consider a regularization Ω_v which controls not only the sparsity but also the structure of the supports of dictionary elements.

2.2 Structured sparsity-inducing norms

The work of [1] considered a norm which induces structured sparsity in the following sense: the solutions to a learning problem regularized by this norm have a sparse support which moreover belongs to a certain set of groups of variables. Interesting sets of possible supports include set of variables forming rectangles when arranged on a grid and more generally convex subsets[2].

The framework of [1] can be summarized as follows: if we denote by $\mathcal{G}$ a subset of the power set of $\{1, \dots, p\}$, such that $\bigcup_{G \in \mathcal{G}} G = \{1, \dots, p\}$, we define a norm Ω on a vector $y \in \mathbb{R}^p$ as

$$\Omega(y) = \sum_{G \in \mathcal{G}} \left\{ \sum_{j \in G} (d_j^G)^2 |y_j|^2 \right\}^{\frac{1}{2}} = \sum_{G \in \mathcal{G}} \left\| d^G \circ y \right\|_2,$$

where $(d^G)_{G \in \mathcal{G}} \in \mathbb{R}^{p \times |\mathcal{G}|}$ is a $|\mathcal{G}|$-tuple of p-dimensional vectors such that $d_j^G > 0$ if $j \in G$ and $d_j^G = 0$ otherwise. This norm Ω linearly combines the ℓ_2 norms of possibly overlapping groups of variables, with variables in each group being weighted by $(d^G)_{G \in \mathcal{G}}$. Note that a same variable y_j belonging to two different groups $G_1, G_2 \in \mathcal{G}$ is allowed to be weighted differently in G_1 and G_2 (by respectively $d_j^{G_1}$ and $d_j^{G_2}$).

For specific choices of $\mathcal{G}$, Ω leads to standard sparsity-inducing norms. For example, when $\mathcal{G}$ is the set of all singletons, Ω is the usual ℓ_1 norm (assuming that all the weights are equal to 1).

[1] From [14], we know that our formulation is also equivalent to two unconstrained problems, with the penalizations $\frac{\lambda}{2} \sum_{k=1}^{r} [\Omega_v(V^k)]^2 + [\Omega_u(U^k)]^2$ or $\lambda \sum_{k=1}^{r} \Omega_v(V^k) \Omega_u(U^k)$.

[2] We use the term *convex* informally here. It can however be made precise with the notion of convex subgraphs ([15]).

We focus on the case of a 2-dimensional grid where the set of groups $\mathcal{G}$ is the set of all horizontal and vertical half-spaces (see Fig. 1 taken from [1]). As proved in [1, Theorem 3.1], the ℓ_1/ℓ_2 norm Ω sets to zero some groups of variables $\|d^G \circ y\|_2$, i.e., some entire horizontal and vertical half-spaces of the grid, and therefore induces rectangular nonzero patterns. Note that a broader set of convex patterns can be obtained by adding in $\mathcal{G}$ half-planes with other orientations. In practice, we use planes with angles which are multiples of $\frac{\pi}{4}$.

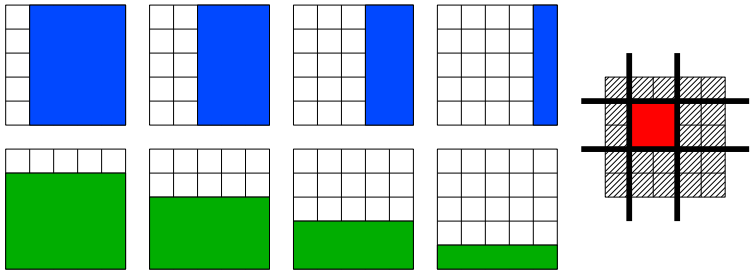

Figure 1: (Left) The set of blue and green groups with their (not displayed) complements to penalize to select rectangles. (Right) In red, an example of recovered pattern in this setting.

Among sparsity inducing regularizations, ℓ_1 is often privileged since it is convex. However, so-called concave penalizations, such as penalization by an ℓ_α quasi-norm, which are closer to ℓ_0 and penalize more aggressively small coefficients can be preferred, especially in a context where the unregularized problem, here dictionary learning is itself non convex. In light of recent work showing the advantages of addressing sparse regression problems through concave penalization (e.g., see [16]), we therefore generalize Ω to a family of non-convex regularizers as follows: for $\alpha \in (0, 1)$, we define the quasi-norm Ω^α for all vectors $y \in \mathbb{R}^p$ as

$$\Omega^\alpha(y) = \left\{ \sum_{G \in \mathcal{G}} \|d^G \circ y\|_2^\alpha \right\}^{\frac{1}{\alpha}} = \| \left(\|d^G \circ y\|_2\right)_{G \in \mathcal{G}} \|_\alpha ,$$

where we denote by $\left(\|d^G \circ y\|_2\right)_{G \in \mathcal{G}} \in \mathbb{R}^{1 \times |\mathcal{G}|}$ the $|\mathcal{G}|$-tuple composed of the different blocks $\|d^G \circ y\|_2$. We thus replace the (convex) ℓ_1/ℓ_2 norm Ω by the (neither convex, nor concave) ℓ_α/ℓ_2 quasi-norm Ω^α. Note that this modification impacts the sparsity induced at the level of groups, since we have replaced the convex ℓ_1 norm by the concave ℓ_α quasi-norm.

3 Optimization

We consider the optimization of Eq. (1) where we use $\Omega_v = \Omega^\alpha$ to regularize the dictionary V. We discuss in Section 3.3 which norms Ω_u we can handle in this optimization framework.

3.1 Formulation as a sequence of convex problems

We are now considering Eq. (1) where we take Ω_v to be Ω^α, that is,

$$\min_{U \in \mathbb{R}^{n \times r}, V \in \mathbb{R}^{p \times r}} \frac{1}{2np} \left\| X - UV^\top \right\|_{\mathrm{F}}^2 + \lambda \sum_{k=1}^{r} \Omega^\alpha(V^k) \qquad \text{s.t.} \qquad \forall k,\, \Omega_u(U^k) \leq 1. \tag{2}$$

Although the minimization problem Eq. (2) is still convex in U for V fixed, the converse is not true anymore because of Ω^α. Indeed, the formulation in V is non-differentiable and non-convex. To

address this problem, we use the variational equality based on the following lemma that is related[3] to ideas from [17, 18]:

Lemma 3.1. *Let $\alpha \in (0,2)$ and $\beta = \frac{\alpha}{2-\alpha}$. For any vector $y \in \mathbb{R}^p$, we have the following equality*

$$\|y\|_\alpha = \min_{z \in \mathbb{R}^p_+} \frac{1}{2} \sum_{j=1}^{p} \frac{y_j^2}{z_j} + \frac{1}{2} \|z\|_\beta,$$

and the minimum is uniquely attained for $z_j = |y_j|^{2-\alpha} \|y\|_\alpha^{\alpha-1}$, $\forall j \in \{1, \dots, p\}$.

Proof. Let $\psi : z \mapsto \sum_{j=1}^p y_j^2 z_j^{-1} + \|z\|_\beta$ be the continuously differentiable function defined on $(0, +\infty)$. We have $\lim_{\|z\|_\beta \to \infty} \psi(z) = +\infty$ and $\lim_{z_j \to 0} \psi(z) = +\infty$ if $y_j \neq 0$ (for $y_j = 0$, note that $\min_{z \geq 0} \psi(z) = \min_{z \geq 0, z_j = 0} \psi(z)$). Thus, the infimum exists and it is attained. Taking the derivative w.r.t. z_j (for $z_j > 0$) leads to the expression of the unique minimum, expression that still holds for $z_j = 0$. □

To reformulate problem (2), let us consider the $|\mathcal{G}|$-tuple $(\eta^G)_{G \in \mathcal{G}} \in \mathbb{R}^{r \times |\mathcal{G}|}$ of r-dimensional vectors η^G that satisfy for all $k \in \{1, \dots, r\}$ and $G \in \mathcal{G}$, $\eta_k^G \geq 0$. It follows from Lemma (3.1) that

$$2 \sum_{k=1}^{r} \Omega^\alpha(V^k) = \min_{(\eta^G)_{G \in \mathcal{G}} \in \mathbb{R}_+^{r \times |\mathcal{G}|}} \sum_{k=1}^{r} \left[\|(\eta_k^G)_{G \in \mathcal{G}}\|_\beta + \sum_{G \in \mathcal{G}} \left\| V^k \circ d^G \right\|_2^2 (\eta_k^G)^{-1} \right].$$

If we introduce the matrix $\zeta \in \mathbb{R}^{p \times r}$ defined by[4] $\zeta_{jk} = \left\{ \sum_{G \in \mathcal{G},\, G \ni j} (d_j^G)^2 (\eta_k^G)^{-1} \right\}^{-1}$, we then obtain

$$2 \sum_{k=1}^{r} \Omega^\alpha(V^k) = \min_{(\eta^G)_{G \in \mathcal{G}} \in \mathbb{R}_+^{r \times |\mathcal{G}|}} \sum_{k=1}^{r} (V^k)^\top \mathrm{Diag}\left(\zeta^k\right)^{-1} V^k + \|(\eta_k^G)_{G \in \mathcal{G}}\|_\beta.$$

This leads to the following formulation

$$\min_{\substack{U,\, V,\, \Omega_u(U^k) \leq 1 \\ (\eta^G)_{G \in \mathcal{G}} \in \mathbb{R}_+^{r \times |\mathcal{G}|}}} \frac{1}{2np} \left\| X - UV^\top \right\|_{\mathrm{F}}^2 + \frac{\lambda}{2} \sum_{k=1}^{r} \left[(V^k)^\top \mathrm{Diag}\left(\zeta^k\right)^{-1} V^k + \|(\eta_k^G)_{G \in \mathcal{G}}\|_\beta \right], \quad (3)$$

which is equivalent to Eq. (2) and convex with respect to V.

3.2 Sharing structure among dictionary elements

So far, the regularization quasi-norm Ω^α has been used to induce a structure *inside* each dictionary element taken separately. Nonetheless, some applications may also benefit from a control of the structure *across* dictionary elements. For instance it can be desirable to impose the constraint that r dictionary elements share only a few different nonzero patterns. In the context of face recognition, this could be relevant to model the variability of faces as the combined variability of several parts, with each part having a small support (such as eyes), and having its variance itself explained by *several* dictionary elements (corresponding for example to the color of the eyes).

To this end, we consider $\mathcal{M}$, a partition of $\{1, \dots, r\}$. Imposing that two dictionary elements V^k and $V^{k'}$ share the same sparsity pattern is equivalent to imposing that V_i^k and $V_i^{k'}$ are simultaneously zero or non-zero. Following the approach used for joint feature selection ([19])

[3] Note that we depart form [17, 18] who consider a quadratic upperbound on the *squared* norm. We prefer to remain in the standard dictionary learning framework where the penalization is not squared.

[4] For the sake of clarity, we do not specify the dependence of ζ on $(\eta^G)_{G \in \mathcal{G}}$.

where the ℓ_1 norm is composed with an ℓ_2 norm, we compose the norm Ω^α with the ℓ_2 norm $V_i^M = \|(V_i^k)_{k\in M}\|_2$, of all i^{th} entries of each dictionary element of a class M of the partition, leading to the regularization:

$$\sum_{M\in\mathcal{M}} \Omega^\alpha(V_i^M) = \sum_{M\in\mathcal{M}} \left[\sum_{G\in\mathcal{G}} \left\|(V_i^k d_i^G)_{i\in G,\, k\in M}\right\|_2^\alpha\right]^{1/\alpha}, \tag{4}$$

In fact, not surprisingly given that similar results hold for the group Lasso [17], it can be shown that the above extension is equivalent to the variational formulation

$$\min_{\substack{U,\, V,\, \Omega_u(U^k)\le 1\\ (\eta^G)_{G\in\mathcal{G}}\in\mathbb{R}_+^{|\mathcal{M}|\times|\mathcal{G}|}}} \frac{1}{2np}\left\|X - UV^\top\right\|_{\mathrm{F}}^2 \;+\; \frac{\lambda}{2}\sum_{M\in\mathcal{M}}\left[\sum_{k\in M}(V^k)^\top \mathrm{Diag}\big(\zeta^M\big)^{-1}V^k + \|(\eta_M^G)_{G\in\mathcal{G}}\|_\beta\right]$$

with class specific variables η_M, ζ^M, $M\in\mathcal{M}$, defined by analogy with η_k and ζ^k, $k\in\{1,\ldots,r\}$.

3.3 Algorithm

The main optimization procedure described in Algorithm 1 is based on a cyclic optimization over the three variables involved, namely $(\eta^G)_{G\in\mathcal{G}}$, U and V. We use Lemma (3.1) to solve Eq. (2) by a sequence of problems that are convex in U for fixed V (and conversely, convex in V for fixed U). For this sequence of problems, we then present efficient optimization procedures based on block coordinate descent (BCD) [20, Section 2.7]. We describe these in detail in Algorithm 1. Note that we depart from the approach of [1] who use an active set algorithm. Their approach does not indeed allow warm restarts, which is crucial in our alternating optimization scheme.

Update of $(\eta^G)_{G\in\mathcal{G}}$ The update of $(\eta^G)_{G\in\mathcal{G}}$ is straightforward (even if the underlying minimization problem is non-convex), since the minimizer $(\eta^G)^*$ in Lemma (3.1) is given in closed-form. In practice, as in [18], we avoid numerical instabilities near zero with the smoothed update $\eta_k^G \leftarrow (\eta_k^G)^* + \varepsilon$, with $\varepsilon \ll 1$.

Update of U The update of U follows the technique suggested by [11]. Each column U^k of U is constrained separately through $\Omega_u(U^k)$. Furthermore, if we assume that V and $\{U^j\}_{j\neq k}$ are fixed, some basic algebra leads to

$$\begin{aligned}\operatorname*{arg\,min}_{\Omega_u(U^k)\le 1} \frac{1}{2np}\left\|X - UV^\top\right\|_{\mathrm{F}}^2 &= \operatorname*{arg\,min}_{\Omega_u(U^k)\le 1} \left\|U^k - \left\|V^k\right\|_2^{-2}(X - \sum_{j\neq k}[U^j]^\top V^j)V^k\right\|_2^2 &(5)\\ &= \operatorname*{arg\,min}_{\Omega_u(U^k)\le 1} \left\|U^k - w\right\|_2^2, &(6)\end{aligned}$$

which is simply the Euclidian projection $\Pi_{\Omega_u}(w)$ of w onto the unit ball of Ω_u. Consequently, the cost of the BCD update of U depends on how fast we can perform this projection; the ℓ_1 and ℓ_2 norms are typical cases where the projection can be computed efficiently. In the experiments, we take Ω_u to be the ℓ_2 norm.

In addition, since the function $U^k \mapsto \frac{1}{2np}\left\|X - UV^\top\right\|_{\mathrm{F}}^2$ is continuously differentiable on the (closed convex) unit ball of Ω_u, the convergence of the BCD procedure is guaranteed since the minimum in Eq. (5) is unique [20, Proposition 2.7.1]. The complete update of U is given in Algorithm 1.

Update of V A fairly natural way to update V would be to compute the closed form solutions available for each row of V. Indeed, both the loss $\frac{1}{2np}\left\|X-UV^\top\right\|_{\mathrm{F}}^2$ and the penalization on V are separable in the rows of V, leading to p independent ridge-regression problems, implying in turn p matrix inversions.

However, in light of the update of U, we consider again a BCD scheme on the columns of V that turns out to be much more efficient, without requiring any non-diagonal matrix inversion. The detailed procedure is given in Algorithm 1. The convergence follows along the same arguments as those used for U.

Algorithm 1 Main optimization procedure for solving Eq. (3).

Input: Dictionary size r, data matrix X.
Initialization: Random initialization of U, V.
  while (*stopping criterion* not reached)
    Update $(\eta^G)_{G\in\mathcal{G}}$: closed-form solution given by Lemma (3.1).
    Update U by BCD:
      for $t = 1$ **to** T_u**, for** $k = 1$ **to** r**:**
        $U^k \leftarrow \Pi_{\Omega_u}(U^k + \left\|V^k\right\|_2^{-2}(XV^k - UV^\top V^k))$.
    Update V by BCD:
      for $t = 1$ **to** T_v**, for** $k = 1$ **to** r**:**
        $V^k \leftarrow \mathrm{Diag}\left(\zeta^k\right)\mathrm{Diag}\left(\left\|U^k\right\|_2^2\zeta^k + np\lambda\mathbf{1}\right)^{-1}(X^\top U^k - VU^\top U^k + \left\|U^k\right\|_2^2 V^k)$.
Output: Decomposition U, V.

Our problem is not *jointly* convex in $(\eta^G)_{G\in\mathcal{G}}$, U and V, which raises the question of the sensitivity of the optimization to its initialization. This point will be discussed in the experiments, Section 4. In practice, the stopping criterion relies on the relative decrease (typically 10^{-3}) in the cost function in Eq. (2).

Algorithmic complexity The complexity of Algorithm 1 can be decomposed into 3 terms, corresponding to the update procedures of $(\eta^G)_{G\in\mathcal{G}}$, U and V. We denote by T_u (respectively T_v) the number of updates of U (respectively V) in Algorithm 1. First, computing $(\eta^G)_{G\in\mathcal{G}}$ and ζ costs $O(r|\mathcal{G}| + |\mathcal{G}|\sum_{G\in\mathcal{G}}|G| + r\sum_{j=1}^p|G\in\mathcal{G}; G\ni j|) = O(pr|\mathcal{G}| + p|\mathcal{G}|^2)$. The update of U requires $O((p+T_u n)r^2 + (np + C_\Pi T_u)r)$ operations, where C_Π is the cost of projecting onto the unit ball of Ω_u. Similarly, we get for the update of V a complexity of $O((n+T_v p)r^2 + npr)$. In practice, we notice that the BCD updates for both U and V require only few steps, so that we choose $T_u = T_v = 3$. In our experiments, the algorithmic complexity simplifies to $O(p^2 + r^2\max\{n,p\} + rp\max\{p^{1/2}, n\})$ times the number of iterations in Algorithm 1.

Extension to NMF Our formalism does not cover the positivity constraints of non-negative matrix factorization, but it is straightforward to extend it at the cost of an additional threshold operation (to project onto the positive orthant) in the BCD updates of U and V.

4 Experiments

We first focus on the application of SSPCA to a face recognition problem and we show that, by adding a sparse structured prior instead of a simple sparse prior, we gain in robustness to occlusions. We then apply SSPCA to biological data to study the dynamics of a protein/protein complex.

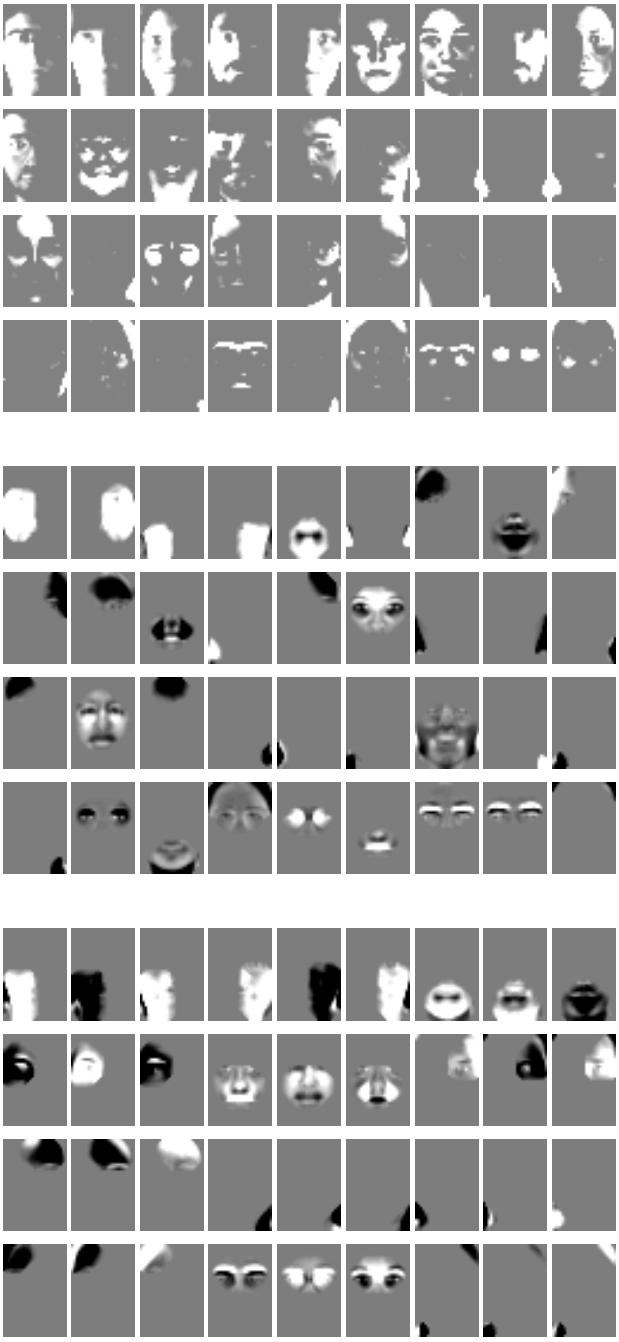

Figure 2: Three learned dictionaries of faces with $r = 36$: NMF (top), SSPCA (middle) and shared-SSPCA (bottom) (i.e., SSPCA with $|\mathcal{M}| = 12$ different patterns of size 3). The dictionary elements are sorted in decreasing order of variance explained. While NMF gives sparse spatially unconstrained patterns, SSPCA finds convex areas that correspond to more natural face segments. SSPCA captures the left/right illuminations in the dataset by recovering symmetric patterns.

The results we obtain are validated by known properties of the complex. In preliminary experiments, we considered the exact regularization of [1], i.e., with $\alpha = 1$, but found that the obtained

patterns were not sufficiently sparse and salient. We therefore turned to the setting where the parameter α is in $(0, 1)$. In the experiments described in this section we chose $\alpha = 0.5$.

4.1 Face recognition

We first apply SSPCA on the cropped AR Face Database [21] that consists of 2600 face images, corresponding to 100 individuals (50 women and 50 men). For each subject, there are 14 non-occluded poses and 12 occluded ones (the occlusions are due to sunglasses and scarfs). We reduce the resolution of the images from 165x120 to 38x27 for computational reasons.

Fig. 2 shows examples of learned dictionaries (for $r = 36$ elements), for NMF, SSPCA and SSPCA with shared structure. While NMF finds sparse but spatially unconstrained patterns, SSPCA select sparse convex areas that correspond to a more natural segment of faces. For instance, meaningful parts such as the mouth and the eyes are recovered by the dictionary.

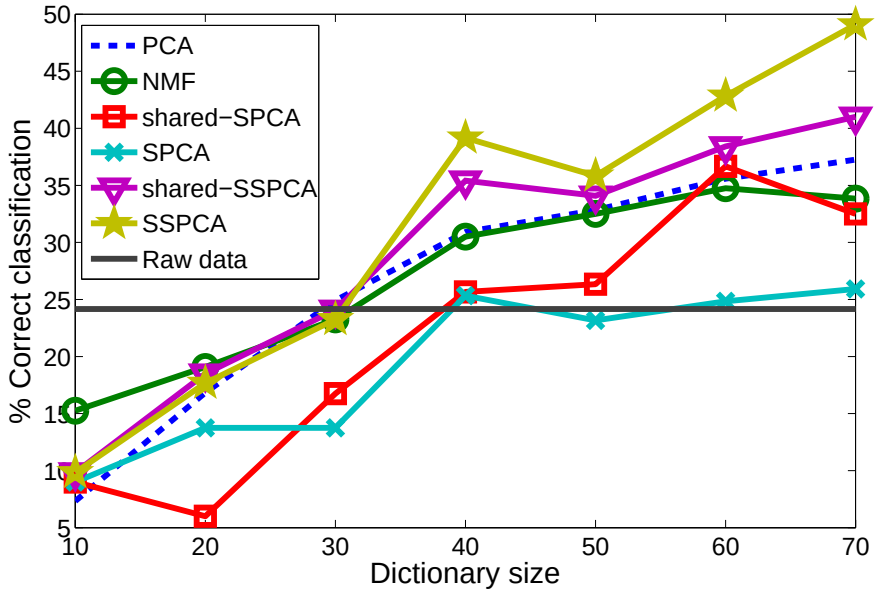

Figure 3: Correct classification rate vs. dictionary size: each dimensionality reduction technique is used with k-NN to classify occluded faces. SSPCA shows better robustness to occlusions.

We now compare SSPCA, SPCA (as in [12]), PCA and NMF on a face recognition problem. We first split the data into 2 parts, the occluded faces and non-occluded ones. For different sizes of the dictionary, we apply each of the aforementioned dimensionality reduction techniques to the non-occluded faces. Keeping the learned dictionary V, we decompose both non-occluded and occluded faces on V. We then classify the occluded faces with a k-nearest-neighbors classifier (k-NN), based on the obtained low-dimensional representations. Given the size of the dictionary, we choose the number of nearest neighbor(s) and the amount of regularization λ by 5-fold cross-validation[5].

The formulations of NMF, SPCA and SSPCA are non-convex and as a consequence, the local minima reached by those methods are sensitive to the initialization. Thus, after having selected the

[5]In the 5-fold cross-validation, the number of nearest neighbor(s) is searched in $\{1, 3, 5\}$ while $\log_2(\lambda)$ is in $\{4, 6, 8, \ldots, 18\}$. For the dictionary, we consider the sizes $r \in \{10, 20, 30, 40, 50, 60, 70\}$.

parameters by cross-validation, we run each algorithm 20 times with different initializations on the non-occluded faces, divided into a training (900 instances) and validation set (500 instances) and take the model with the best classification score. We summarize the results in Fig. 3. We denote by shared-SSPCA (resp. shared-SPCA) the models where we impose, on top of the structure of Ω^α, to have only 10 different nonzero patterns among the learned dictionaries (see Section 3.2).

As a baseline, we also plot the classification score that we obtain when we directly apply k-NN on the raw data, without preprocessing. Because of its local dictionary, SSPCA proves to be more robust to occlusions and therefore outperforms the other methods on this classification task. On the other hand, SPCA, that yields sparsity without a structure prior, performs poorly. Sharing structure across the dictionary elements (see Section 3.2) seems to help SPCA for which no structure information is otherwise available.

The goal of our paper is not to compete with state-of-the-art techniques of face recognition, but to demonstrate the improvement obtained between ℓ_1 and more structured norms. We could still improve upon our results using non-linear classification (e.g., with a SVM) or by refining our features (e.g., with a Laplacian filter).

4.2 Protein complex dynamics

Understanding the dynamics of protein complexes is important since conformational changes of the complex are responsible for modification of the biological function of the proteins in the complex. For the EF-CAM complex we consider, it is of particular interest to study the interaction between EF (adenyl cyclase of *Bacillus anthracis*) and CAM (calmodulin) to find new ways to block the action of anthrax [22].

In our experiment, we consider 12000 successive positions of the 619 residues of the EF-CAM complex, each residue being represented by its position in $\mathbb{R}^3$ (i.e., a total of 1857 variables). We look for dictionary elements that explain the dynamics of the complex, with the constraint that these dictionary elements have to be small convex regions in space. Indeed, the complex is comprised of several functional domains (see Fig. 4) whose spatial structure has to be preserved by our decomposition.

We use the norm Ω^α (and $\mathcal{G}$) to take into account the spatial configuration. Thus, we naturally extend the groups designed for a 2-dimensional grid (see Fig. 1) to a 3-dimensional setting. Since one residue corresponds to 3 variables, we could either (1) aggregate these variables into a single one and consider a 619-dimensional problem, or (2) we could use Section 3.2 to force, for each residue, the decompositions of all three coordinates to share the same support, i.e., in a 1857-dimensional problem. This second method has given us more satisfactory results.

We only present results on a small dictionary (see Fig. 4 with $r = 3$). As a heuristic to pick λ, we maximize $|\bigcup_{k=1}^r \mathrm{supp}(V^k)|^2/(p\sum_{k=1}^r |\mathrm{supp}(V^k)|)$ to select dictionary elements that cover pretty well the complex, without too many overlapping areas.

We retrieve groups of residues that match known energetically stable substructures of the complex [22] . In particular, we recover the two tails of the complex and the interface between EF and CAM where the two proteins bind. Finally, we also run our method on the same EF-CAM complex perturbed by (2 and 4) calcium elements. Interestingly, we observe stable decompositions, which is in agreement with the analysis of [22].

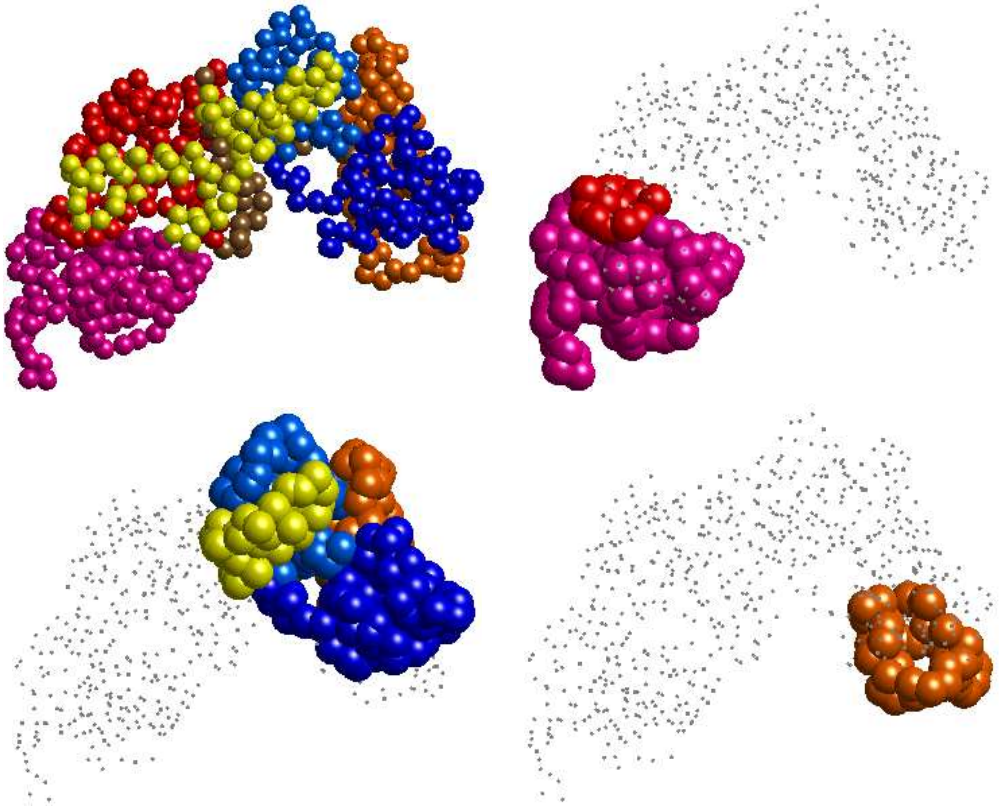

Figure 4: (Top left) Entire protein with biological domains highlighted in different colors. The two blue parts represent the CAM protein, while the rest of the complex corresponds to EF. (Top right, bottom left/right) Dictionary of size $r = 3$ found by SSPCA with the same color code.

5 Conclusions

We proposed to apply a non-convex variant of the regularization introduced by [1] to the problem of structured sparse dictionary learning. We present an efficient block-coordinate descent algorithm with closed-form updates. For face recognition, the dictionaries learned have increased robustness to occlusions compared to NMF. An application to the analysis of protein complexes reveals biologically meaningful structures of the complex. As future directions, we plan to refine our optimization scheme to better exploit sparsity. We also intend to apply this structured sparsity-inducing norm for multi-task learning, in order to take advantage of the structure between tasks.

Acknowledgments

We would like to thank Elodie Laine, Arnaud Blondel and Thérèse Malliavin from the Unité de Bioinformatique Structurale, URA CNRS 2185 at Institut Pasteur, Paris for proposing the analysis of the EF-CAM complex and fruitful discussions on protein complex dynamics. We also thank Julien Mairal for sharing his insights on dictionary learning.

References

[1] R. Jenatton, J-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. Technical report, 2009. preprint arXiv:0904.3523v1.

[2] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[3] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin. A modified principal component technique based on the LASSO. *Journal of Computational and Graphical Statistics*, 12(3):531–547, 2003.

[4] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

[5] D. M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics (Oxford, England)*, April 2009.

[6] L. Jacob, G. Obozinski, and J. P. Vert. Group Lasso with overlap and graph Lasso. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

[7] J. Huang, T. Zhang, and D. Metaxas. Learning with structured sparsity. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

[8] A. d'Aspremont, F. Bach, and L. El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9:1269–1294, 2008.

[9] B. Moghaddam, Y. Weiss, and S. Avidan. Spectral bounds for sparse PCA: Exact and greedy algorithms. *Advances in Neural Information Processing Systems*, 18:915, 2006.

[10] Lester Mackey. Deflation methods for sparse pca. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1017–1024. 2009.

[11] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

[12] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, pages 801–808, 2007.

[13] A.P. Singh and G.J. Gordon. A Unified View of Matrix Factorization Models. In *Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases-Part II*, pages 358–373. Springer, 2008.

[14] F. Bach, J. Mairal, and J. Ponce. Convex sparse matrix factorizations. Technical report, 2008. preprint arXiv:0812.1869.

[15] F.R.K. Chung. *Spectral graph theory*. American Mathematical Society, 1997.

[16] T. Zhang. Multi-stage convex relaxation for learning with sparse regularization. In *Advances in Neural Information Processing Systems*, pages 1929–1936, 2008.

[17] F. Bach. Consistency of the group Lasso and multiple kernel learning. *Journal of Machine Learning Research*, 9:1179–1225, 2008.

[18] C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6(2):1099, 2006.

[19] G. Obozinski, B. Taskar, and M. Jordan. Joint Covariate Selection and Joint Subspace Selection for Multiple Classification Problems. *Statistics and Computing*, 2009.

[20] D. P. Bertsekas. *Nonlinear programming*. Athena scientific, 1995.

[21] A. M. Martinez and A. C. Kak. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[22] E. Laine, A. Blondel, and T. E. Malliavin. Dynamics and energetics: A consensus analysis of the impact of calcium on ef-cam protein complex. *Biophysical Journal*, 96(4):1249–1263, 2009.